

# A piramis

**Készítette:**

Név: Varga Bence Gyula

Neptun-azonosító: QFFOK2

Gyakorlatvezető neve: Tóth Zoltán

## Tartalom

Felhasználói dokumentáció.....	3
Feladat.....	3
Környezet.....	3
Minimum rendszerkövetelmény .....	3
Szükséges szoftverkövetelmény.....	3
Használata .....	3
A program indítása .....	3
Főmenü.....	3
Irányítás .....	4
Játékmenet .....	4
Fejlesztői dokumentáció.....	4
Feladat.....	4
Környezet.....	4
Projekt felépítése .....	5
Main.py .....	5
Menu.py .....	5
Jatekvezerles.py.....	5
Megjelenites.py .....	5
Fajlkezeles.py.....	5
Ranglista.txt.....	6
Kerdesekek.txt.....	6
Adatszerkezetek .....	6
Függvények.....	7

# Felhasználói dokumentáció

## Feladat

A piramis program arra való, hogy egy szórakoztató kikapcsolódást nyújtson a felhasználónak, ami közben akár új ismeretekre tehet szert. A játék során a felhasználó egy műveltségi vetélkedőn vesz részt és a jó válaszokat, a játék egy különleges módon jutalmazza. A játék felépítése, hogy három alap forduló van, a végén egy bónusz fordulóval. Egy fordulóban három darab kérdés van négy válasz lehetőséggel. Minden forduló végén megjelenik a piramis, melynek alján megnyerhető összegek szerepelnek. A piramis tetején hat darab lyuk található, amelyekből három darab labda fog leesni. Amelyik összeget tartalmazó rekeszbe esik a labda, azt az összeget a program jóváírja a felhasználónak. Viszont ez nem ilyen egyszerű, mivel a piramis belsejében akadályok találhatók, amikről irányt tud változtatni a labda, illetve a piramis nem csak adni tud, de el is vesz. Amennyi kérdésre rosszul válaszolt a játékos, a három közül annyi labda lesz büntető labda. Ugyan úgy leesnek a lyukakból de ezek levonják az összeget. Minden korosztály számára kiváló szórakozás lehet, mivel a kérdések nehézség szerint csoportosítva vannak.

## Környezet

### Minimum rendszerkövetelmény

1. Microsoft Windows 8 vagy későbbi verziója, MacOS 10.13 vagy ennél újabb verzió, Bármilyen Linux verzió ami tartalmazza a Gnome, KDE kiegészítőt.
2. 4GB RAM vagy ennél több
3. A monitor felbontása legalább 1024x768
4. Egér és billentyűzet

### Szükséges szoftverkövetelmény

1. Python 3: 3.5-től 3.9-ig
2. Tkinter könyvtár letöltése
3. Pymunk könyvtár letöltése
4. Pygame könyvtár letöltése
5. A garantált tökéletes működéshez PyCharm fejlesztőkörnyezetben futtatni

## Használata

### A program indítása

A piramis nevű mappa megnyitása, majd az általunk kiválasztott környezetben a main.py fájl elindítása. Fontos, hogy a tkinter és a pymunk könyvtárak telepítése elengedhetetlen a futáshoz.

### Főmenü

Indítás után egyből a főmenüvel találkozik a felhasználó, melyben egérrel tud navigálni. Három részre van bontva a menü:

**Játék indítása:** Ezzel lehet a játékot elindítani. Első körben kilehet választani, hogy milyen nehézségen szeretnénk a játékkal játszani. A nehézség itt a kérdések nehézségi szintjét jelenti más nem befolyásol. Ez után megtudja adni a felhasználó, hogy milyen névvel szeretne játszani, itt minden karakter elfogadható, de csak 14 karakter hosszúságú név megadása lehetséges. Csak akkor enged tovább a játék, ha megadtunk valamilyen nevet ekkor meg is jelenik egy indítás gomb.

**Ranglista:** Ez a menüpont az eddigi legjobb 10 eredményt írja ki. Megjeleníti a ranglistában való helyezést, a játékosnevét, milyen nehézségi szinten játszott a játékos és hogy mennyi összeget nyert meg. A ranglistában a legjobb három helyezés kiemelt színnel rendelkezik. Visszalépés a főmenübe a bal felső nyílra kattintással lehetséges.

**Kilépés:** Ezzel lehet bezárni a játékot, de bezárás előtt megjelenik egy ablak, ami megkérdezi, hogy

biztosan szeretnénk zárni a játékot? Két lehetőség közül lehet választani: igenre kattintva a játék bezárul, nemre kattintva pedig visszatér a program a főmenübe.

### Irányítás

Az egész játék során az egérrel lehet navigálni többek között a menüben és a kérdések válaszolásánál is. A játékosnév megadásánál lehet használni a billentyűzetet. A többi játék részben semmilyen külső perifériával nem lehet befolyásolni a játékot.

### Játékmenet

A főmenüben a játék indítása gombra kattintva elindul az első forduló. Minden sima forduló felépítése a következő: Megjelenik egy ablak 6 darab gombbal melyek 1-től 6-ig vannak számozva itt három számot kell kiválasztani. Ezek a számok jelentik azt, hogy ebben a körben melyik három lyukból essen le a labda. A játék magától tovább lép, ha megtörtént a három szám kiválasztása. Utána megjelenik egy kérdés négy válaszlehetőséggel a négy válaszlehetőség közül csak az egyik jó. A gombokra kattintva lehetséges megjelölni egy válaszlehetőséget, ilyenkor más színnel fog megjelenni. Csak egy válaszlehetőséget lehet megjelölni a program automatikus kiértékeli, hogy jó vagy rossz válasz érkezett. Amennyiben jó volt a válasz, akkor a képernyő zöld színben fog megjelenni, ha rossz akkor pedig piros színben, majd a program magától tovább lép és betölti a következő kérdést. Egy fordulóban három kérdés van, ha vége a három kérdésnek akkor megjelenik a piramis, melynek a felépítése úgy néz ki, hogy a piramis alsó részén helyezkednek el az összegek, a tetején pedig 6 db lyuk van ezek azok a lyukak, amiket a játékos ki tud választani az elején. A piramis belső részében pedig akadályok helyezkednek el, amikről irányt tud változtatni a labda. Három darab labda fog leesni melyeknek két színe lehet, piros ha rosszul válaszolt a játékos az adott kérdésre és zöld, ha jól válaszolt az adott kérdésre. A labda véletlenszerű irányba eshet és ha egyszer beleesik az összegek rekeszébe az színe alapján vagy hozzáadja az összeget a nyereményhez vagy levonja. Abban az esetben, ha az összeg 0 értéken van, akkor mínuszba nem mehet. Még továbbá két ilyen felépítésű kör van. A játék utolsó része a bónuszforduló, de ez csak akkor elérhető, ha az alap fordulók végén az összeg nem nulla. Amennyiben nulla, akkor a játék véget ér és felkerül a felhasználói ranglistára. Ha nem nulla, akkor a felhasználó eltudja dönteni, hogy szeretne-e játszani a bónusz fordulóval, mivel ebben a fordulóban meg lehet duplázni a nyereményt, de el is lehet bukni az egészet. A bónusz forduló felépítése hasonló egy sima labdaeséshez, viszont itt a rekeszekben dupla vagy semmi értékek állnak és ezek a rekeszek aránya a játék során jól válaszolt kérdésektől függ. A bónusz forduló után a játék véget ér és felhasználó felkerül a ranglistára majd bezárul a program.

## Fejlesztői dokumentáció

### Feladat

A program feladata, hogy egy kérdéseket tartalmazó txt fájlból beolvassa és megjelenítse azokat a hozzájuk tartozó válaszlehetőségekkel. Mindemelett grafikus megjelenítést használjon az egész játék során. A menüben az egér segítségével lehet navigálni, majd a játékosnév megadásánál használható a billentyűzet beviteli funkcióra. A játék végén a program képes fájlba kiírni az eredményeket.

### Környezet

A program Python 3 3.9-es verzióban készült Pycharm fejlesztőkörnyezetben ezért a biztos és hibamentes futás csak ilyen környezetben támogatott. Továbbá szükséges három külső könyvtár telepítése az egyik, a Tkinter ami egy válaszablak lehetőséget biztosít. Ez van használva a kilépés menünél, ahol megjelenik egy válaszablak, hogy biztosan akar-e lépni a játékból. Itt két lehetőség közül lehet választani igen vagy nem, igen esetén bezárul a program, nem esetén pedig visszalép a főmenübe. Ezt biztosítja a Tkinter külső könyvtár. Egy másik szükséges külső könyvtár pedig a Pymunk

ez a könyvtár biztosít bizonyos grafikai megjelenítéseket és fizikai kalkulációkat. A harmadik szükséges pedig a Pygame többek között a szövegek és gombok használatát és megjelenítését biztosítja.

### Projekt felépítése

A projekt összesen 5 darab vezérlőfájlt tartalmaz, kettő darab txt állományt és két darab png kiterjesztésű képet.

#### Main.py

Itt kerül inicializálásra a Pygame könyvtár, ami létre hozza a játék ablakát a megadott paraméterekkel: Felbontás állít, ami 1280x720, beállítja a címét az ablaknak, hozzárendeli az ablakhoz a játék logóját és kitölti a képernyőt a menü fő színével. Meghívja a menu.py modult.

#### Menu.py

Ez a modul felel a játék főmenü kezelésére és megjelenítésére. Első sorban meghív külső könyvtárakat: Pygame és Tkinter. Három nagyobb csoportra bontható, a játékindítása ezzel lehet elindítani a játékot és egészen a név megadásáig ez a modul felelős mindenért. Amint megadott a felhasználó egy játékosnevet és rákattint az indítás gombra akkor meghívja a játékvezérlés modult. A ranglista menüpont a ranglista.txt fájlt olvassa be és ennek segítségével jeleníti meg az eredményeket. Fontos, hogy a ranglista.txt fájl olvasva és írva is lesz a program futása során. A harmadik nagyobb elem pedig a kilépés, ha erre kattint a felhasználó akkor egy párbeszéd ablak jelenik meg. A program megkérdezi a felhasználót, hogy biztosan beakarja-e zárni a játékot, ekkor igen és nem opció közül tud választani. Itt jön használatban a Tkinter külső könyvtár.

#### Jatekvezerles.py

A modul feladata, hogy kezelje a játékmenetét akkor kerül használatba amikor a felhasználó a játékindítása mellett dönt és már választott nehézséget, megadott egy játékosnevet. Ebben a modulban történik a legfontosabb adatok feldolgozása ezért is tartalmaz két osztályt, melyekről az osztályok pontban lehet bővebben olvasni. Többek között ez a modul hívja meg a fájlkezelés modult a kérdések beolvasása miatt és ezeket a kérdéseket nehézség szerint csoportosítja is. Itt történik a kérdések közül véletlen kérdés kiválasztása is, akkor tárolja a játéksorán hány kérdésre válaszolt jól a felhasználó. Számon tartja folyamatosan, hogy egy fordulóban milyen interakciókat csinált a felhasználó, melyik számokat választotta a számválasztásnál és hogy a kérdésekre jó vagy rossz válaszokat adott. Kommunikál a megjelenítés modullal.

#### Megjelenites.py

A modul főfeladata, hogy minden grafikai elemet megjelenítsen és kezelje is a felhasználó által érkező bemeneteket. Használja a külső Pygame és Pymunk könyvtárat. Ennek a modulnak segítségével jelenik meg a számválasztás, minden átvezető szöveg, a kérdések megjelenítése, a piramis felépítése és annak megjelenítése.

#### Fajlkezeles.py

Ez a modul kommunikál és kezeli a külső txt fájlokat. Többek között beolvassa az összes kérdést egy txt fájlból és továbbítja azokat a játékvezérlés modulnak. Képes a ranglista.txt fájlból olvasni adatokat, amiket továbbít a menü modulnak, illetve a játékvezérlés a játék végén meghívja a fájlkezelés modult és adatokat ír a ranglista.txt-be. Minden fájl használatnál van hibakezelés azokra az esetekre, ha a megadott fájl nem lenne megtalálható és minden fájl UTF-8 kódolással van használva.

### Ranglista.txt

A ranglistához tartozó adatokat tárolja kettősponttal elválasztva ilyen sorrendben: A játékosneve, nehézségi szint, amin játszott és végül az elvitt összeg. Minden egy játékoshoz tartozó adatt egy sorban van, tehát soronként vannak egy játékos adatai. Ez a fájl olvasva és írva is van a program futása során.

### Kerdések.txt

Több mint 5000 kérdés adatait tárolja, soronként egy kérdés adatait pedig tabulátorral választja el. Egy kérdés adatait pedig a következők: Nehézség ez egy 1-től 14-ig terjedő szám, maga a kérdés majd ezt követi négy válasz lehetőség utána pedig, hogy melyik válasz volt jó a négy közül. Az utolsó oszlopban pedig a kérdés témaköre van.

### Adatszerkezetek

A program működésének igazán két főszereplője van. Az egyik szereplő a játékos adatok, ami tárolja a felhasználó adatait a játék során. Eltárolja a nevét, a nehézségi szintjét és hogy mennyi az összeg, amit elvitt. A másik szereplő pedig a kérdések adatok ez az adatszerkezet tárolja a beérkező kérdések adatait. Többek között a kérdések nehézségi szintjét, amiket számként kap meg és besorolja őket három nagyobb csoportra. Eltárolja maga a kérdést a négy válasz lehetőséggel, azt is eltárolja, hogy a kérdésre melyik a jó válasz, illetve a kérdés kategóriáját is számon tartja, de ezt a jelenlegi program nem használja fel, csak a további fejleszthetőség szempontjából érdemes.

```
class jatekos_adatok:
    def __init__(self,nev,neheszseg,osszeg):
        self.nev = nev
        self.nehezseg = neheszseg
        self.osszeg = int(osszeg)
```

Ez egy játékos adatait tároló osztály a játékvezérlés modulban található és ez a modul is gyűjti össze a benne tárolható adatokat. A játékvezérlés osztály paraméterként megkapja a játékos nevét és hogy milyen nehézségi szintet választott. Az összeg kezdetben nulla értékű. Ez az osztály lista felépítésre szolgál, ha vége a játéknak a játékvezérlés modul egy listát épít a játékos adatok osztályt felhasználva és tovább küldi ezt a listát a fájlkezelésnek, ahol már könnyedén hivatkozva ki lehet írni a ranglistába az adatokat. Ez az osztály a ranglista építésben is segít még pedig a fájlkezelés modul hívja meg és egy olyan listát épít fel, ami jatekos\_adatok osztályból állnak.

```
class kerdes_adatok:
    def __init__(self,neheszseg,kerdes,A,B,C,D,valasz,kategoria):
        if neheszseg >=1 and neheszseg <4:
            self.nehezseg = "könnyű"
        elif neheszseg >=4 and neheszseg <9:
            self.nehezseg = "közepes"
        elif neheszseg >=9:
            self.nehezseg = "nehéz"
        self.kerdes = kerdes
        self.A = str(A)
        self.B = str(B)
        self.C = str(C)
        self.D = str(D)
        self.valasz = valasz
        self.kategoria = kategoria
```

Ez az osztály egy kérdésnek az adatait tárolja a fájlkezelés modul hívja meg. Mivel a kérdések nehézségi szintje egy 1-től 14-ig tartó skálán mozognak az egyszerűbb használhatóság érdekében három csoportra bontja értékek szerint. Minden adatott stringként tárol.

## Függvények

`main.menu()`

Ez a függvény építi fel a program alapjait többek között létrehoz egy ablakot 1280x720-as felbontásban, beállítja az ablak címét, beállítja a játék logóját és kitölti egy alapszínnel a képernyőt. Majd meghívja a `menu.main_menu(ablak)` függvényt aminek paraméterként továbbítja a létrehozott ablakot.

`menu.main_menu(ablak)`

Ez a függvény készíti el a főmenüt és a hozzátartozó rész menüket. További függvényeket hív meg, illetve paraméterként kap egy ablakot, hogy hol kell megjeleníteni az elemeket. Az egér mozgását is érzékeli és ha egy menüelem fölé visszük az egeret, akkor más színnel jeleníti meg az aktuális szöveget. Innen három függvényhívás történhet. Abban az esetben ha a felhasználó a játékindítás gombra kattint, akkor a `menu.jatek_inditasa(ablak)` függvényt hívja meg. Ha a felhasználó a ranglista gombra kattint, akkor pedig a `menu.ranglista(ablak)` függvényt hívja meg. Végül pedig, ha a kilépés gombra kattint, akkor a kilépés függvényt kerül meghívásra.

`menu.jatek_inditasa(ablak)`

Ha a menüben a játékos megnyomja a játék indítása gombot, akkor hívja meg a program ezt a függvényt és itt lehet kiválasztani milyen nehézségen szeretnénk a játékkal játszani. Paraméterként kap egy ablakot ahová a szövegeket írja. Végül meghívja a `jatekos_nev_megadasa` függvényt, ami ha visszatért egy játékosnévvel akkor meghívja a `jatekvezerles.main(ablak,nehhezseg,jatekosnev)` függvényt. Kezeli az egér mozgást és ha valamelyik gomb fölé visszük az egeret, akkor más színnel fog megjeleníteni.

`menu.jatekos_nev_megadasa(ablak)`

Nehézség kiválasztása után hívja meg a program. Tartalmazza a játékosnév megadására alkalmas szövegdobozt, melybe csak akkor lehet írni, ha a felhasználó rákattint. Itt mindenféle karakter elfogadható kivéve a space, tabulátor és az enter. A játékosnév maximum hossza pedig 14 karakter lehet. Ha a játékos megadott valamilyen nevet akkor megjelenik egy indítás gomb arra kattintva visszatér a `menu.jatek_inditasa(ablak)` függvénybe és visszaadja a felhasználó által beírt nevet.

`menu.ranglista(ablak)`

Az egész ranglista megjelenítéséért felelős. A `menu.main_menu(ablak)` függvény hívja meg és ez a függvény kettő további függvényt hív meg `menu.ranglista_sorszamok_kiirasa(ablak)` és a `menu.ranglista_helyezesek_kiirasa(ablak)` függvényt.

`menu.ranglista_sorszamok_kiirasa(ablak)`

A ranglista sorszámokat írja ki 1-től 10-ig a kapott paraméter ablakra. Az első három sorszámot más színnel jeleníti meg, mint a többi. A `menu.ranglista(ablak)` függvény hívja meg

`menu.ranglista_helyezesek_kiirasa(ablak)`

Ez a függvény meghívja a `fajlkezeles.ranglista_beolvas()` függvényt ami egy listával tér vissza ami `jatekos_adatok` osztályokból állnak. Végig megy a listán és kiírja a listában szereplő elemeket

`menu.kilepes()`

A `main_menu(ablak)` függvény hívja meg és egy párbeszéd ablakot indít a felhasználóval, hogy biztosan kiseretne-e lépni a játékból itt igen és nem közül lehet választani, ha igen akkor kilép a játékból és bezárja az ablakot, ha pedig nem akkor visszatér a `main.menu(ablak)` függvénybe.

menu.szoveg\_kiiras(betutipus,meret,szoveg,szin)

Ez a függvény az összes szöveg megjelenítésben részt vesz, ha a játék során bármilyen szöveggel találkozik a felhasználó akkor ez a függvény lesz meghívva. Paraméterként kap egy betűtípust, méretet, maga a szöveget, hogy mit írjon ki és hogy milyen színnel tegye azt. Visszatérési értéként csak egy szöveg típusú elemet fog visszaadni.

jatekvezlerles.main(ablak,nehhezseg,jatekosnev)

A menu.jatek\_inditasa(ablak) függvény hívja meg. Ez kezeli az egész játék menetét, írja ki a játék részeit (átvezető szöveg, számválasztás, kérdések megjelenítése, labdaesés, további függvények segítségével beolvassa a kérdéseket és kiírja a ranglistába az eredményt). Nyilván tartja felhasználó jó válaszainak számát, hogy szeretne-e bónuszfordulót, éppen jelenleg mennyi a játékos összege, ezek nyilvántartására listákat és int típusú változókat használ. Függvényeket hív meg a fájlkezelés és a megjelenítése modulból is. Paraméterként megkapja az ablakot a játékos nevét és a választott nehézséget ezeket fogja a ranglistába beleírni.

jatekvezlerles.kerdesek\_nehezseg\_szerint(lista,nehhezseg)

Paraméterként megkapja az összes kérdést egy listában és megkapja, hogy a felhasználó milyen nehézségi szinten játszik. Végig megy az összes kérdésen és amelyik kérdésnek olyan nehézségi szintje mint amit választott a felhasználó, azokat kigyűjti egy másik listába és visszaadja a main függvénynek.

jatekvezlerles.veletlen\_kerdes(lista)

Már csak azokat a kérdéseket kapja meg amiket az előző függvény kiválogatott, és azok a kérdések közül kiválaszt véletlenszerűen egyet, majd visszatér vele a main függvénybe.

jatekvezlerles.jatekos\_valasztott\_szamok(lista)

Paraméterként egy true-false elemekből álló listát kap. A true érték azt jelenti, hogy azon az indexen levő számot választotta a játékos. Végig megy ezen a listán és visszaad egy másik listát melyben már konkrétan számként vannak a választott számok.

jatekvezlerles.labdak\_szine(lista)

Egy listát kap paraméterként ez a lista azt tárolja, hogy a felhasználó a három kérdés közül, melyikre válaszolt jól és melyikre rosszul ezt az információt felhasználva végig megy a listán és egy másik listában visszaadja, hogy milyen színűek legyenek a labdák. Ha jól válaszolt a játékos akkor „Green” érték kerül a listába, ha pedig rosszul akkor meg „Red” érték kerül.

jatekvezlerles.jo\_valaszok\_szama(lista)

Listát kap paraméterként, ami tárolja a válaszok eredményét. Végig megy ezen a listán és megszámlolja, hogy hány jó kérdése lett a felhasználónak. Visszatérési értéként ezt a darabszámot adja vissza.

jatekvezlerles.ranglista(jatekosnev,nehhezseg,osszeg)

Paraméterként megkapja a játékosnevét, választott nehézségét és hogy mennyi összeggel fejezte be a játékot a kapott információkból előállít egy jatekos\_adatok osztályból álló listát és meghívja a fajlekezeles.ranglista\_kiir(lista) függvényt.

megjelenites.szam\_valasztas(ablak)

Ez a függvény a számválasztás játékrész megjelenítését és annak kezelését végzi. Paraméterként megkapja, hogy melyik ablakra kell megjeleníteni a gombokat, szövegeket. Magja egy ciklus, ami addig fut amíg a felhasználó rá nem kattint három darab számra, illetve létrehoz egy választva tömböt kezdetben tele false értékekkel, amit visszatérési értéként fog visszaadni. Figyeli, hogy a



játékos melyik számokat választotta és amelyiket igen azt a gombot inaktívvá teszi és a választva listában az megfelelő indexen átírja True-ra az értéket. Ha megtörtént a három számválasztás akkor visszatér a `jatekvezertes.main()` függvénybe a választott tömbbel.

`megjelenites.atvezeto_szoveg(ablak,szoveg,x,y)`

Ez a függvény a játékban megtalálható átv vezető szövegeinek megjelenítéséért felelős. Paraméterként megkapja, hogy melyik ablakra kell a szöveget írni, illetve, hogy mi maga a szöveg, amit kisseretnének írni és milyen pozícióban helyezkedjen el.

`megjelenites.lesz_bonusz_fordulo(ablak)`

Ha a játék végén a felhasználó összeg nagyobb, mint nulla akkor lehetősége van eldönteni, hogy szeretne-e a bónuszfordulóval játszani. Ez a függvény kiírja a képernyőre, a szöveget és egy igen nem gombot jelenít meg. Ha a felhasználó a gombok fölé viszi az egeret akkor más színnel fog megjeleníteni. Majd ha a felhasználó rákattint az egyik gombra, akkor visszatér a `fajlkezeles.main()` függvénybe. Két visszatérési értéke lehet. Abban az esetben, ha a játékos az igen gombra nyomott akkor true értékkel tér vissza, abban az esetben, ha a játékos a nem gombra kattintott akkor pedig false értékkel.

`megjelenites.jatek_vege_szoveg(ablak,osszeg)`

Paraméterként megkapja a játékos végső összegét és azt, hogy melyik ablakra írja ki a szövegeket. A függvény kiírja, hogy a felhasználó milyen összeggel zárta a játékot, kiír egy elköszönő üzenetet és kiírja a játékkészítőjét.

`megjelenites.egy_kerdes(ablak,adatok)`

A függvény megjelenít és kezel egy teljes kérdés kiírását a hozzátartozó válaszokkal figyeli, hogy a felhasználó melyik válaszlehetőséget jelölte meg és eredménytől függően megváltoztatja az ablak színét. Ha jól válaszolt a felhasználó a kérdésre akkor zöld lesz az ablak, ha rosszul akkor pedig piros. Visszatérési értéként True vagy False eredménnyel tér vissza a `jatekvezertes.main` függvénybe. A program addig marad ebben a függvényben, amíg a felhasználó nem választ egy választási lehetőséget.

`megjelenites.akadaly_test(jatekter,x,y)`

A piramis belsejében elhelyezkedő akadályok fizikai vázát hozza létre a Pymunk könyvtár segítségével, ehhez paraméterként megkapja, hogy melyik játéktérhez adja hozzá és hogy hol helyezkedjen el. Visszatérési értéként pedig visszaadja a fizikai vázát.

`megjelenites.akadalyok_megjelenitese()`

Létrehoz egy tömböt amiben az összes akadályt eltárolja melyhez felhasználja a `megjelenites.akadaly_test()` függvényt visszatérési értéként ezt a tömböt adja.

`megjelenites.akadaly_rajzol(ablak,akadalyok)`

Paraméterként egy listát ad amiben az akadályok fizikai vázai találhatóak. Végig megy ezen a listán és minden akadály fizikai vázához elkészíti a grafikai megjelenést.

`megjelenites.labda_test(jatekter,x,y)`

A piramisban használt labda fizikai vázát készíti el a pymunk könyvtár segítségével. Paraméterként megkapja, hogy hova helyezze el a labda vázát és hogy melyik játéktérre.

`megjelenites.labda_rajzol(ablak,labda,szin)`

A piramisban használt labda grafikai részét valósítja meg. Ehhez paraméterként megkapja, hogy melyik ablakon jelenítse meg a labdát, megkapja a fizikai vázát a labdának és hogy milyen színben rajzolja meg a labdát. A szín pedig attól függ, hogy a felhasználó milyen eredménnyel zárta a kérdést

megjelenites.falak\_rajzol(ablak,jatekter)

Ez a függvény a piramis alap elemeit készíti el, a falakat és a megnyerhető összegek rekeszeit. Paraméterként pedig egy ablakot kap, hogy melyik ablakra jelenítse meg grafikai részét és egy játékkeret kap, hogy hol helyezze el az elemek fizikai vázát.

megjelenites.lyukak\_rajzol(ablak)

A függvény megjeleníti a piramis tetején elhelyezkedő lyukakat, innen esnek le a labdák pontosan 6 darab lyukat jelenít meg és paraméterként megkapja, hogy melyik ablakon jelenítse meg.

megjelenites\_nyeremenyek\_kiir(ablak,kor)

Ez a függvény tárol egy listát amiben három lista szerepel. A három lista pedig 9 darab különféle összeget tartalmaz. Majd a paraméterben kapott ablakra megjeleníti az összeget és azt az indexű listát használja fel, amit paraméterként kapott. Ezek az összegek szereplenek a piramisban a rekeszek alatt.

megjelenites.labdaeses(ablak,szinek,labdak,kor,osszeg)

Ez a függvény jeleníti és kezeli a labdaesés játékrészt. Paraméterként megkapja az ablakot, hogy hol kell megjeleníteni az elemeket, megkapja a labdák színeit és hogy azok a labdák melyik lyukakból essenek le, illetve számon tartja, hogy a felhasználó éppen melyik körben jár és hogy mennyi az aktuális összege. Ezt az összeget ez a függvény növeli vagy csökkenti. Használja a Pymunk külső könyvtárát és létrehoz egy játékkeret, beállítja a gravitáció szintjét. Egy mátrixban tárolja a körönként a megnyerhető összegeket és felépíti a teljes játékot. Meghívja a megjelenites.lyukak\_rajzol(), megjelenites.falak\_rajzol(), megjelenites.akadaly\_rajzol(), megjelenites.labda\_rajzol() függvényeket. Visszatérési értékben visszaadja, hogy a felhasználó mennyi összeggel rendelkezik az adott kör végén.

megjelenites.bonusz\_eredmenyek\_kiir(ablak,eredmenyek)

Paraméterként kap egy összegeket tartalmazó listát. Ebben a listában dupla vagy semmi string változók vannak és ezeknek az aránya egy másik függvény segítségével jön létre. Végig megy a listán és kiírja az eredményeket az adott ablakra.

megjelenites.bonusz\_nyeremenyek\_lista(jovalaszok)

Paraméterként megkapja, hogy a felhasználó mennyi kérdésre válaszolt jól a játék során és ez a változó alapján elkészít egy listát. Ez a lista dupla vagy semmi stringeket tartalmaz és ezek aránya a jó válaszok számától függ. Ezt a listát visszatérésiértékként adja vissza.

megjelenites.bonusz\_fordulo(ablak,jovalaszok,osszeg)

Ez a függvény a bónuszforduló játék elemet építi fel és kezeli az eseményeket. A sima játéktérhez hasonlóan, több függvény meghívással megjeleníti a piramist, csak ebben az esetben a labda színe és pozíciója nem függ a felhasználótól, illetve a nyereményeket tartalmazó rekeszek alján dupla vagy semmi szövegeket tartalmaz. Ha a labda semmit tartalmazó rekeszbe esik, akkor az eddigi összeget lenullázza. Ha pedig egy dupla szövegeket tartalmazó rekeszbe esik, akkor megkétszerezi az aktuális összeget, majd visszatér a jatekvezelres.main() függvénybe a játékos összegével.

fajlkezeles.ranglista\_beolvas()

Megnyitja a ranglista.txt fájlt, ami tartalmazza az eddigi ranglistában szereplő eredményeket. Végig megy a fájl sorain és visszatérésiértékként készít egy listát, ami játékos\_adatok osztályból áll, majd visszaadja ezt a listát. Kivételt dob, ha nem találja a txt fájlt.

`fajlkezeles.ranglista_kiir(lista)`

Paraméterként megkap egy listát ami tartalmazza az aktuális felhasználó adatait. A játékosnevét, nehézségi szintet amin játszott, és hogy mennyi összeggel zárta a játékot. Majd a ranglista.txt fájlba beleírja ezeket az adatokat. Kivételt dob, ha nem találja a fájlt.

`fajlkezeles.kerdesek_beolvas()`

Megnyitja a kerdesek.txt fájlt és a sorait beolvassa. Ez a fájl tartalmazza a játék során használt kérdéseket. Az adatokat felhasználva épít egy `kerdesek_adatok` osztályt tartalmazó listát és ezt a listát visszatérési értékben visszaadja.